curvature of each of particles corresponding to the vertices and a distance between the vertices and the particles.

[0142] In operation **1625**, the expression apparatus controls an application intensity of the selected caustic textures. For example, the expression apparatus controls an application intensity of a caustic texture based on an angle between the surface normal vector of the first object and an incident direction of rays.

[0143] In operation **1630**, the expression apparatus adjusts an application size of the selected caustic textures. For example, the expression apparatus adjusts an application size of a caustic texture based on a distance between positions of surface particles of the first object and positions of the vertices.

[0144] In operation **1635**, the expression apparatus places virtual particles on a caustic map based on a complexity of particles corresponding to the caustic textures with the controlled application intensity and the adjusted application size.

[0145] In operation **1640**, the expression apparatus generates, based on the virtual particles, caustic lines used to connect the caustic textures with the controlled application intensity and the adjusted application size to each other.

[0146] In operation **1645**, the expression apparatus generates the caustic map by connecting the caustic textures to each other by the caustic lines.

[0147] In operation **1650**, the expression apparatus expresses the first object by combining the caustic map with a specular map and a diffuse map.

[0148] FIG. **17** illustrates an an expression apparatus **1700** in accordance with one or more embodiments. Referring to FIG. **17**, the expression apparatus **1700** includes a processor **1710**, a memory **1720**, and a receiver **1730**. The processor **1710**, the memory **1720** and the receiver **1730** communicate with each other via a bus **1740**.

[0149] The processor **1710** calculates intersection positions at which rays emitted from a light source pass through particles of a first object and meet a second object, and expresses the first object using a caustic map generated by applying caustic textures to the intersection positions.

[0150] The memory **1720** stores one or more caustic textures.

[0151] The receiver **1730** receives at least one information among: a position of the light source, a depth map pre-rendered from the position of the light source, positions of the particles, and a surface normal vector of the first object. The processor **1710** calculates the intersection positions for each of the particles based on the information received by the receiver **1730**.

[0152] The processor **1710** determines particles directly visible from the light source among the particles and calculates intersection positions for the particles directly visible from the light source.

[0153] Also, the processor **1710** controls an application intensity of a caustic texture based on an angle between the surface normal vector of the first object and an incident direction of the rays. The processor **1710** adjusts an application size of a caustic texture based on a distance between positions of surface particles of the first object and intersection positions corresponding to the surface particles.

[0154] The processor **1710** marks vertices corresponding to the intersection positions, and applies the caustic textures to the vertices.

[0155] The processor **1710** selects caustic textures based on at least one of a fluid surface curvature of each of particles corresponding to the vertices and a distance between the vertices and the particles, and places the selected caustic textures on the caustic map to correspond to the vertices.

[0156] The processor **1710** places virtual particles on the caustic map based on a complexity of particles corresponding to the selected caustic textures, generates, based on the virtual particles, caustic lines used to connect the selected caustic textures to each other, and connects the selected caustic textures to each other by the caustic lines.

[0157] In addition, the processor **1710** performs at least one of the operations described above with reference to FIGS. **1** through **16**. The processor **1710** executes a program and controls the expression apparatus **1700**. A program code executed by the processor **1710** is stored in the memory **1720**. The expression apparatus **1700** is connected to an external apparatus, for example, a personal computer (PC) or a network, via an input/output device (not shown), and exchanges data with the external apparatus.

[0158] At least one of the operations described above with reference to FIGS. **1** through **16** is used in combination with software to edit a three-dimensional (3D) image or a 3D graphic image. Also, at least one of the operations described above with reference to FIGS. **1** through **16** is implemented in the form of an application operating in a processor included in a tablet or a smartphone, or in the form of a chip, and is included in a display.

[0159] FIG. **18** illustrates an example of an operation of a processor **1800** in an expression apparatus in accordance with one or more embodiments. Referring to FIG. **18**, the processor **1800** includes the vertex shader **1810** and a fragment shader **1820**.

[0160] The processor **1800** determines whether rays and a second object collide with each other, for each particle, and expresses caustics using the vertex shader **1810** and/or the fragment shader **1820** at least once.

[0161] The processor **1800** receives, as inputs, information, for example, positions of particles, a surface normal vector of the first object, a position of a light source and/or a depth map pre-rendered from the position of the light source.

[0162] The vertex shader **1810** calculates intersection positions based on the received information. The vertex shader **1810** calculates intersection positions with a neighboring second object (for example, a bottom surface) for particles that directly receives light, that is, particles directly visible from a light source, among particles viewed from a position of a camera. Whether a ray directly reaches a particle is determined, in one or more embodiments, by comparing depth information of a depth map rendered from a position of the light source and depth information of a particle viewed from a camera in a space of the light source.

[0163] The fragment shader **1820** generates a caustic map by combining caustic textures in the intersection positions received from the vertex shader **1810**, and transfers the caustic map to the processor **1800**. The processor **1800** combines the caustic map with another rendering element and generates a final first object rendering image.

[0164] According to one or more embodiments, to perform calculation to locate intersection positions for each pixel, a fragment shader, for example, may be used instead of a vertex shader. Even though the fragment shader calcu-